

**ANL252 (Online)**

**PYTHON FOR DATA ANALYTICS**

**Tutor-Marked Assignment**

**July 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **DARRION GOH ING HEAN** | **E2310655** |

**Tutorial Group: ­­­­­­­­­­­­ T 03**

**Instructor’s Name: Mr. MUNISH KUMAR**

**Submission Date: 14/09/2023**

Q1a)

Why Plagiarism in Coding Happens:

**Pressure and Deadlines:** Pressure to keep up with their grades and tight project timelines may lead to shortcuts and copying code. (Bailey. J, 2020)

**Lack of Understanding:** Some developers may not fully comprehend the code they're using, leading to unintentional plagiarism.

**Inexperience:** Novice programmers may resort to plagiarism due to limited knowledge. (“Why do Students Plagiarize”,2021)

**Competition:** A desire to outperform peers or gain recognition can push individuals towards plagiarism.

How to Avoid Plagiarism in Coding:

**Understand the Code:** Ensure a deep understanding of the code you're using.

**Cite Sources:** Properly credit and cite any borrowed code, libraries, or frameworks. (George. T, 2021)

**Document Your Work:** Maintain clear documentation of your code, explaining its purpose and functionality.

**Continuous Learning:** Invest in improving your coding skills and knowledge to reduce reliance on plagiarism.

By plagiarizing or cheating on assignments, students unfairly give themselves an advantage and disrespect other students' efforts. The best way for students to mitigate the risks associated with plagiarism is to follow these practices, fostering a culture of integrity and innovation in their coding practices.

(183 words)

Q1b)

car\_brands = ("Toyota", "Kia", "Honda")

print (f"This is our list of car brands: {car\_brands}" )

lentuple = len(car\_brands)

user\_qn = str(input("Whcih car brand do you want: "))

car\_count = 0

for j in range(lentuple):

if car\_brands[j] == user\_qn:

count\_car = j

break

if car\_brands[count\_car] == user\_qn:

print(f"You have entered a valid car brand {car\_brands[count\_car]} ")

else:

print(f"Error")

(Kumar. M, 2023)

Firstly, we put three different car brands into a tuple named “car\_brands”. Using the “print” function, it will show the list of car brands available to the user. We use the “lentuple” function to return the number of elements there are in the tuple (tutorialspoint). We call the “user\_qn” so that the user can input the car brand that he or she wants, and the input would be stored in the “user\_qn” variable. The “car\_brand” tuple is then iterated through using a for loop after I initialize the “car\_count” variable to zero. Within the for loop, it then checks each element in the tuple if it matches the user’s input (user\_qn). Afterwards, it is necessary to break out of the loop as soon as there is a match. It then stores the index of that match in the “count\_car” variable. Finally, the code checks if the car brand stored at “count\_car” in the “car\_brands” tuple matches the user input. If it matches, it will print that the user has entered a valid car brand. However, if there is no match, it will print an error message instead.

(187 words)

Q1c) **Code:**

cars = ("Toyota", "Kia", "Honda", "Nissan")

price = ("$100000", "$120000", "$130000", "$125000")

payment\_methods = ("Visa", "Nets", "Mastercard")

print(f"Here are our lists of cars available: {cars}")

while True:

while True:

car\_input = input("What car do you want to buy? ")

index = -1 # Initialize with a value that indicates "not found"

for i, car in enumerate(cars):

if car.lower() == car\_input.lower(): # Case-insensitive comparison

index = i

break

if index != -1:

selected\_price = price[index]

print(f"Price: {selected\_price}")

break # Exit the fruit input loop when a valid car is provided

else:

print("Sorry, that car is not available. Please choose another car.")

print(f"Here is a list of payment methods available: {payment\_methods}")

while True:

payment\_input = input("How do you want to pay? ")

if payment\_input in payment\_methods:

print(f"Payment method: {payment\_input}")

break # Exit the payment input loop when a valid payment method is provided

else:

print("Invalid payment method. Please choose a valid payment method.")

**Output:**

Here are our lists of cars available: ('Toyota', 'Kia', 'Honda', 'Nissan')

What car do you want to buy? Nissan

Price: $125000

Here is a list of payment methods available: ('Visa', 'Nets', 'Mastercard')

How do you want to pay? Bank Transfer

Invalid payment method. Please choose a valid payment method.

How do you want to pay? ***\*User Input\****

There are numerous reasons why we must rewrite our python codes.

The reasons are the following:

* Understand and learn the codes.
* Customize to our convenience and needs.
* Avoiding plagiarism.

Firstly, rewriting the python codes will allow us to understand the codes better and how the codes work. We may not fully understand the logic of the codes if we simply extract the entire source. Rewriting it will force us to dissect and understand each component, variables, and functions. When we write our own codes, we will encounter many errors or unexpected behaviours. Therefore, debugging these issues will understand their causes and come to a solution.

Secondly, rewriting python codes can help us customize to our convenience. When we are coding, there are some specific requirements that we must fulfil. Extracting codes online may not satisfy those requirements and we may need to change it. This will also help us to change some of the variable names so that we can identify which variables is used for which portion so that it would not be so confusing for us.

Thirdly, rewriting python codes will help us avoid plagiarism (Heard. L, 2019). Rewriting it in our own words for coding will help us to avoid plagiarism. Referencing codes from online sources or books need to be included in our codes or reports so that credits can be given to the original programmer.

(230 words)

Q2)

Firstly, we can use a dictionary to store the products and the prices. Using a dictionary is a better and easier way to store both the name of the product and the price of the product instead of storing them on separate lists. Using two parallel lists can result in more error-prone if the order of elements in the list changes if we need to associate the price and product.

Secondly, we should make the input from the user to be non-case sensitive so that the code can handle upper- and lower-case inputs. For example, if the user enters a product name, “LapToP”, the code may not recognise that it is a valid product in the list in “products”.

Lastly, we can check for a valid price in the “float” format. If the user keys in a “string” value when the code asks for the price of the product, the code will detect it being an error and prints an error message and skip the rest of the loop.

(169 words)

**Code:**

products = ['laptop', 'mouse', 'webcam', 'keyboard', 'speaker']

query = 'yes'

product\_prices = {} # Create an empty dictionary to store product prices

print(f'We have a list of products here: {products}.')

while query == 'yes':

item = input("Hello! What do you want to buy? ").lower() # Convert input to lowercase for case-insensitive matching

if item not in products:

print('Wrong product! Please try again.')

continue

price\_of\_item = input(f"How much is the {item} (in SGD)? ")

try:

price = float(price\_of\_item)

except ValueError:

print('Invalid price format. Please enter a valid number for the price.')

continue # Continue the loop to allow the user to try again

product\_prices[item] = float(price\_of\_item) # Store the price in the dictionary as a float

query = input("Would you like to continue? (yes/no)").lower()

print('This is our updated shopping list:')

for item, price in product\_prices.items():

print(f'{item}: {price} SGD')

References:

Jonathan Bailey (2020, July 9). *Why Do Students Commit Plagiarism?* <https://www.plagiarismtoday.com/2020/07/09/why-do-students-commit-plagiarism/>

Fixgerald (2021, September 16). *Why Do Students Plagiarize: Accidental and Deliberate Reasons.* <https://fixgerald.com/blog/why-do-students-plagiarize>

Munish Kumar (2023, September 5). *4\_Example\_Functions.ipynb https://canvas.suss.edu.sg/courses/61624/discussion\_topics/368273*

Tegan George (2021, October 10). *How to Avoid Plagiarism | Tips on Citing Sources.* <https://www.scribbr.com/plagiarism/how-to-avoid-plagiarism/>

Tutorialspoint. *Python Tuple len() Method*. <https://www.tutorialspoint.com/python/tuple_len.htm>

Simplilearn (2023, February 28) *What Is A Dictionary In Python?* <https://www.simplilearn.com/dictionary-in-python-article#:~:text=In%20Python%2C%20dictionaries%20are%20mutable,elements%20using%20the%20methods%20dict.>

Lisa Heard (2019, December 19) *Plagiarism and Programming: How to Code without Plagiarizing* <https://www.itbriefcase.net/how-to-code-without-plagiarizing>